CUNY SPS DATA 621 - CTG5 - HW3

Betsy Rosalen, Gabrielle Bartomeo, Jeremy O'Brien, Lidiia Tronina, Rose Koh

April 10th, 2019

Contents

Table 1: Data Dictionary

VARIABLE	DEFINITION	TYPE
target	whether the crime rate is above the median crime rate (1) or not (0)	response
zn	proportion of residential land zoned for large lots (over 25000 square feet)	predictor
indus	proportion of non-retail business acres per suburb	predictor
chas	a dummy var. for whether the suburb borders the Charles River (1) or not (0)	predictor
nox	nitrogen oxides concentration (parts per 10 million)	predictor
rm	average number of rooms per dwelling	predictor
age	proportion of owner-occupied units built prior to 1940	predictor
dis	weighted mean of distances to five Boston employment centers	predictor
rad	index of accessibility to radial highways	predictor
tax	full-value property-tax rate per $10,000	predictor
ptratio	pupil-teacher ratio by town	predictor
black	1000(B__k - 0.63)^2 where B__k is the proportion of blacks by town	predictor
lstat	lower status of the population (percent)	predictor
medv	median value of owner-occupied homes in $1000s	predictor

1 DATA EXPLORATION

Relocating to a new city or state can be very stressful. In addition to the stress of packing and moving, you may also be nervous about moving to an unfamiliar area. To better understand their new community, some new residents or people interested in moving to a new city choose to review crime statistics in and around their neighborhood. Crime rate may also influence where people choose to live, raise their families and run their businesses; many potential new residents steer clear of cities with higher than average crime rates.

Data was collected in order to predict whether the neighborhood will be at risk for high crime levels. For each neighborhood the response variable, `target`, represents whether the crime rate is above the median crime rate or not. In addition to that 13 predictor variables were collected representing each neighborhood's: proportion of large lots, non-retail business acres, whether or not it borders the Charles River, nitrogen oxides concentration, average number of rooms per dwelling, proportion of owner-occupied units, distances to five Boston employment centers, accessibility to radial highways, property tax rate, pupil-teacher ratio, proportion of African Americans, percent lower status, and median value of homes. The evaluation data contains the same 13 predictor variables and no target variable so it will be impossible to check the accuracy of our predictions from the testing data.

Table 2: Summary statistics

	n	min	mean	median	max	sd
zn	466	0.0000	11.5772532	0.00000	100.0000	23.3646511
indus	466	0.4600	11.1050215	9.69000	27.7400	6.8458549
chas	466	0.0000	0.0708155	0.00000	1.0000	0.2567920
nox	466	0.3890	0.5543105	0.53800	0.8710	0.1166667
rm	466	3.8630	6.2906738	6.21000	8.7800	0.7048513
age	466	2.9000	68.3675966	77.15000	100.0000	28.3213784
dis	466	1.1296	3.7956929	3.19095	12.1265	2.1069496
rad	466	1.0000	9.5300429	5.00000	24.0000	8.6859272
tax	466	187.0000	409.5021459	334.50000	711.0000	167.9000887
ptratio	466	12.6000	18.3984979	18.90000	22.0000	2.1968447
lstat	466	1.7300	12.6314592	11.35000	37.9700	7.1018907
medv	466	5.0000	22.5892704	21.20000	50.0000	9.2396814
target	466	0.0000	0.4914163	0.00000	1.0000	0.5004636

1.1 Summary Statistics

Looking at the Table. 1, we can see that `chas` and `target` are binary variables. 49% of our target variable is coded as 0's indicating that the crime rate is NOT above the median crime rate. There are potential outliers present in `zn`, `lstat`, `medv` and `dis`.

1.2 Shape of Predictor Distributions

Figure. 1 shows that the distribution of most of the variables seems skewed. There are some outliers in the right tail of `tax` , `rad`, `medv`, `lstat`, `dis` and left tail of `ptratio`.

Even more interestingly, for many of the predictor variables the shape of the distribution is significantly different depending on the value of the `target`. For example, `age` is highly left skewed for homes where the crime rate is above the median crime rate (`target` = 1) while the distribution for homes where the crime rate is not above the median the distribution is normal. Other variables with similar differences are `dis`, `indus`, `lstat`, `nox`, `ptratio`, `rad`, and `tax`.

The variable `rad` has a clear separation at about a value of 5 with almost all homes with a `rad` value less than five being in the target group coded 0 and almost all homes with a `rad` value greater than five being in the target group coded 1. Possibly indicating that a transformation into a categorical dummy variable might be desirable. `indus` has a similar separation at a value of about 16, but not as strikingly.

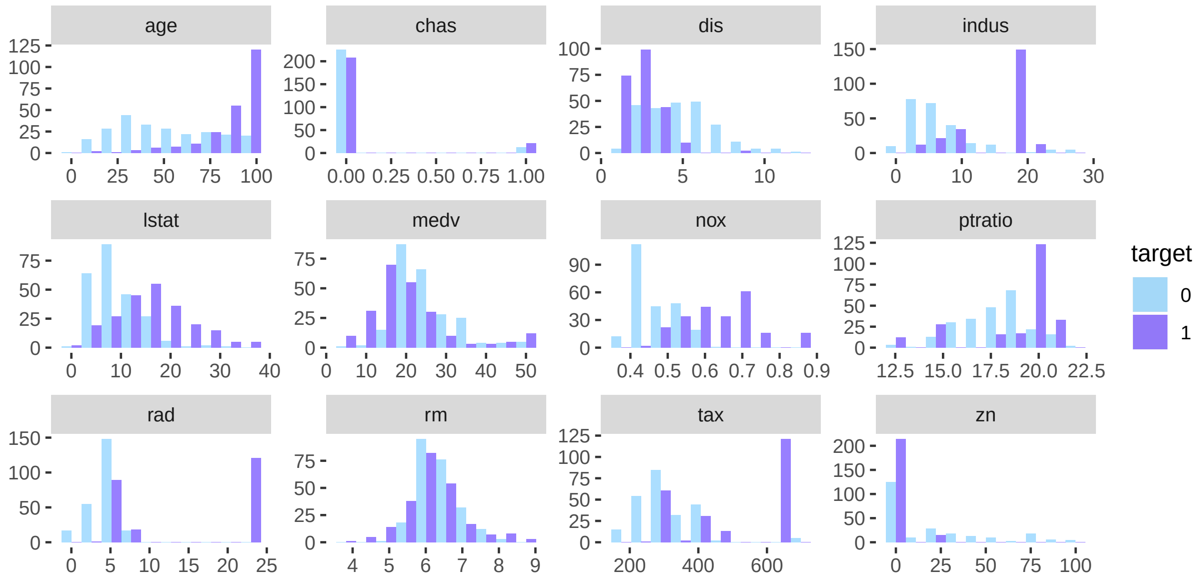


Figure 1: Data Distributions

1.3 Outliers

Figure. 2 shows that there are also a large number of outliers that need to be accounted for, most significantly in `zn` and `medv` and less significantly in `lstat`, `dis` and `rm`. Since `tax` variable has values which are very large compared to other variables in the dataset, it was scaled to fit the boxplot by dividing by 10.

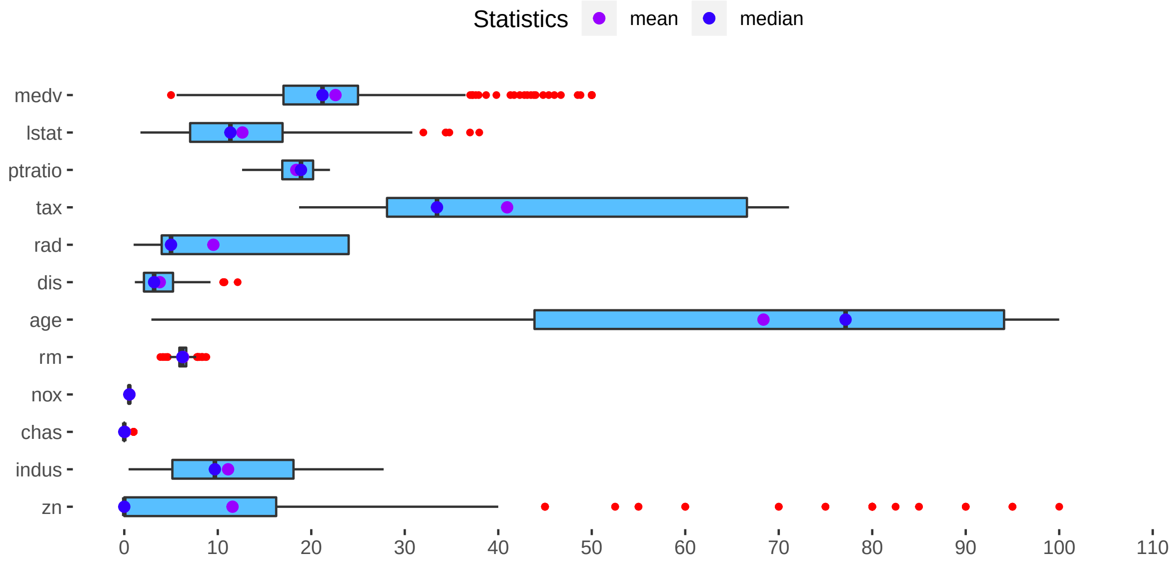


Figure 2: Boxplots highlighting many outliers in the data.

1.4 Missing Values

There are no missing values in any of our observations gathered across the thirteen predictor variables as can be seen in Figure. 3.

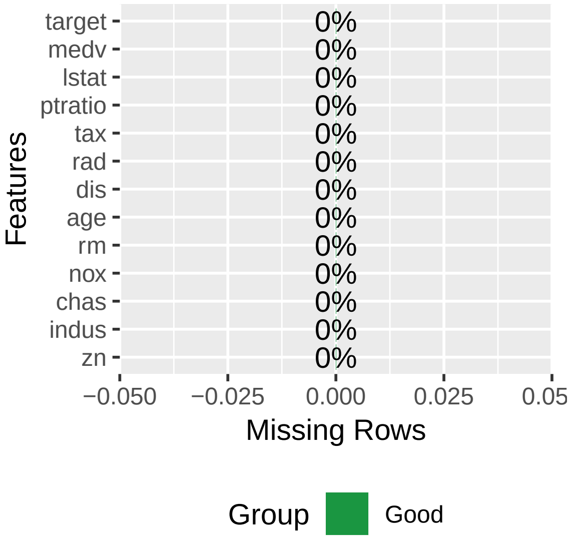


Figure 3: Missing values

1.5 Linearity

Each variable was plotted against the target variable in order to determine at a glance which had the most potential linearity before the dataset was modified.

As can be observed in Figure. 4, all of the predictor variables seem to have an impact on the target. With most of them having a positive impact indicating that the higher the predictor variable values are more likely to correspond to a target that is coded as 1 indicating the crime rate is above the median. The exceptions are `dis`, `medv`, `rm`, `zn`, and possibly `chas` where the distribution of predictor variable values is higher when the target is coded 0.

We can also see that many of the predictor variables have very different variances for the two values of the target. This is especially true for `age`, `rad`, `tax`, and `zn` and less significantly for `dis` and `nox`.

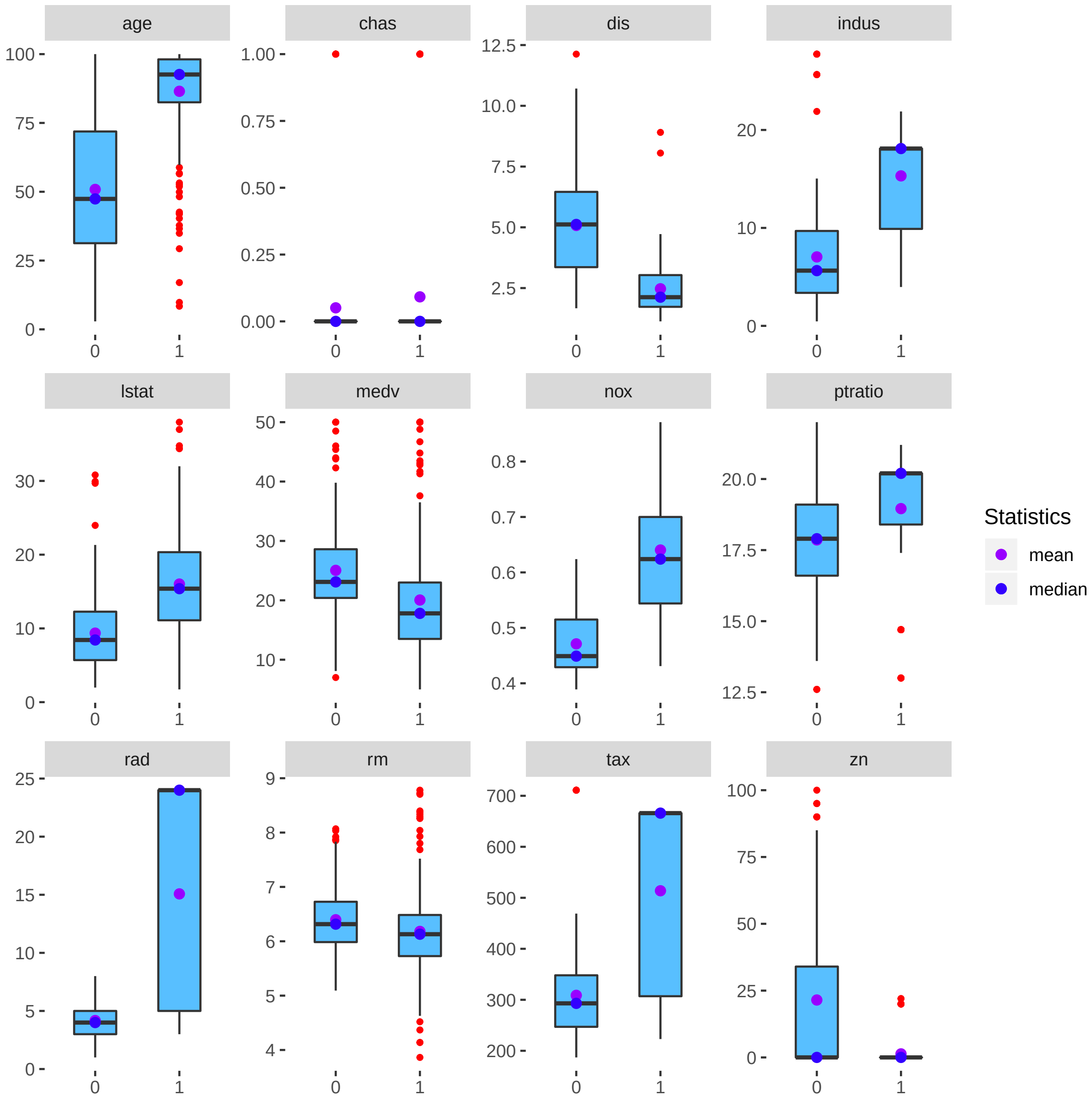


Figure 4: Linear relationships between each predictor and the target

2 DATA PREPARATION

2.1 Missing Values and NA Imputation

Given that the training dataset does include missing values, there's no need to make systematic corrections or imputations.

2.2 Dealing with outliers, leverage, and influence points

While logistic regression can be more robust to leverage points (explanatory variable values, which are distant on the x-axis), outliers (response variable values, which are distant on the y-axis) can exert influence which affects the curve and accuracy of target predictions.

- `dis`, `tax` (property tax rate per $10k), and `medv` (median value of owner-occupied homes) see a few outliers and leverage points in both target classes
- `indus` (the non-retail business acreage proportion) and `lstat` (percent lower status population) both have outliers in the below-mean (0) class
- `ptratio` (pupil-teacher ratio) fit is very impacted by density of low values in the above-mean class, making the linear relationship appear parabolic
- `rad` (highway access index) is influenced by a high-value concentration of locations distant from radial highways that fall in the above-mean class
- `rm` (average rooms per dwelling) sees a wider distribution of house size for the above-mean class then the below-mean; while `zn` (large-lot zoned land proportion) sees the opposite, with a concentration around a few non-residential land proportions for the above-mean class and a wide dispersion for the below-mean class

The figures below examine the linear relationships after a log transformation, which smoothes several relationships but still demonstrates visible influence for several other variables: `lstat`, `medv`, `ptratio`, `rad`, `rm`, `tax`, and `zn`. We discuss further in the feature engineering section below.

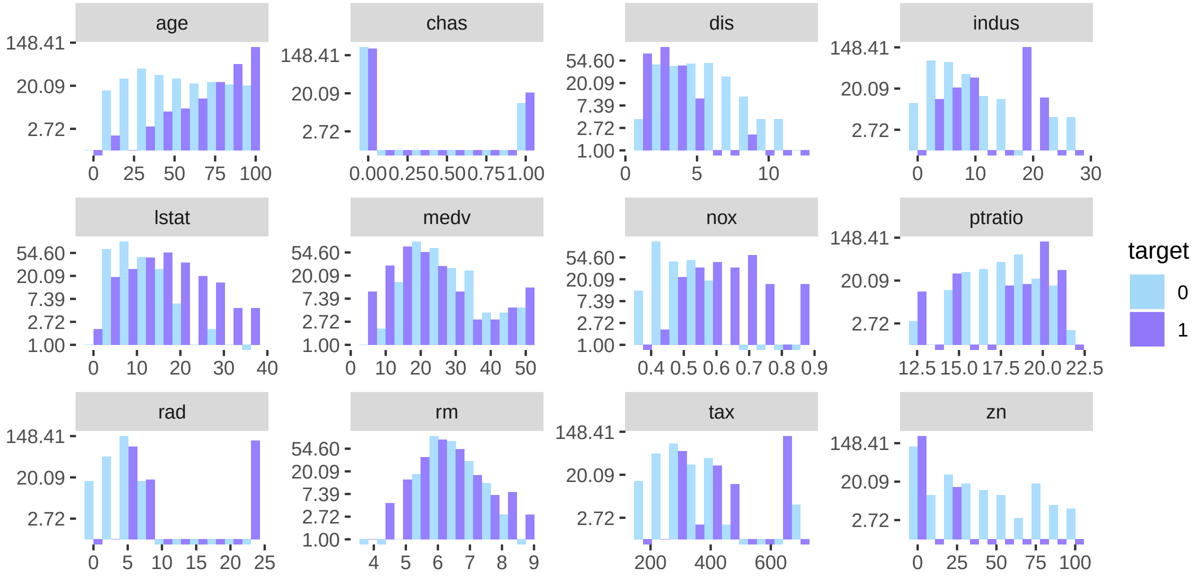


Figure 5: Natural log transformed predictor distributions

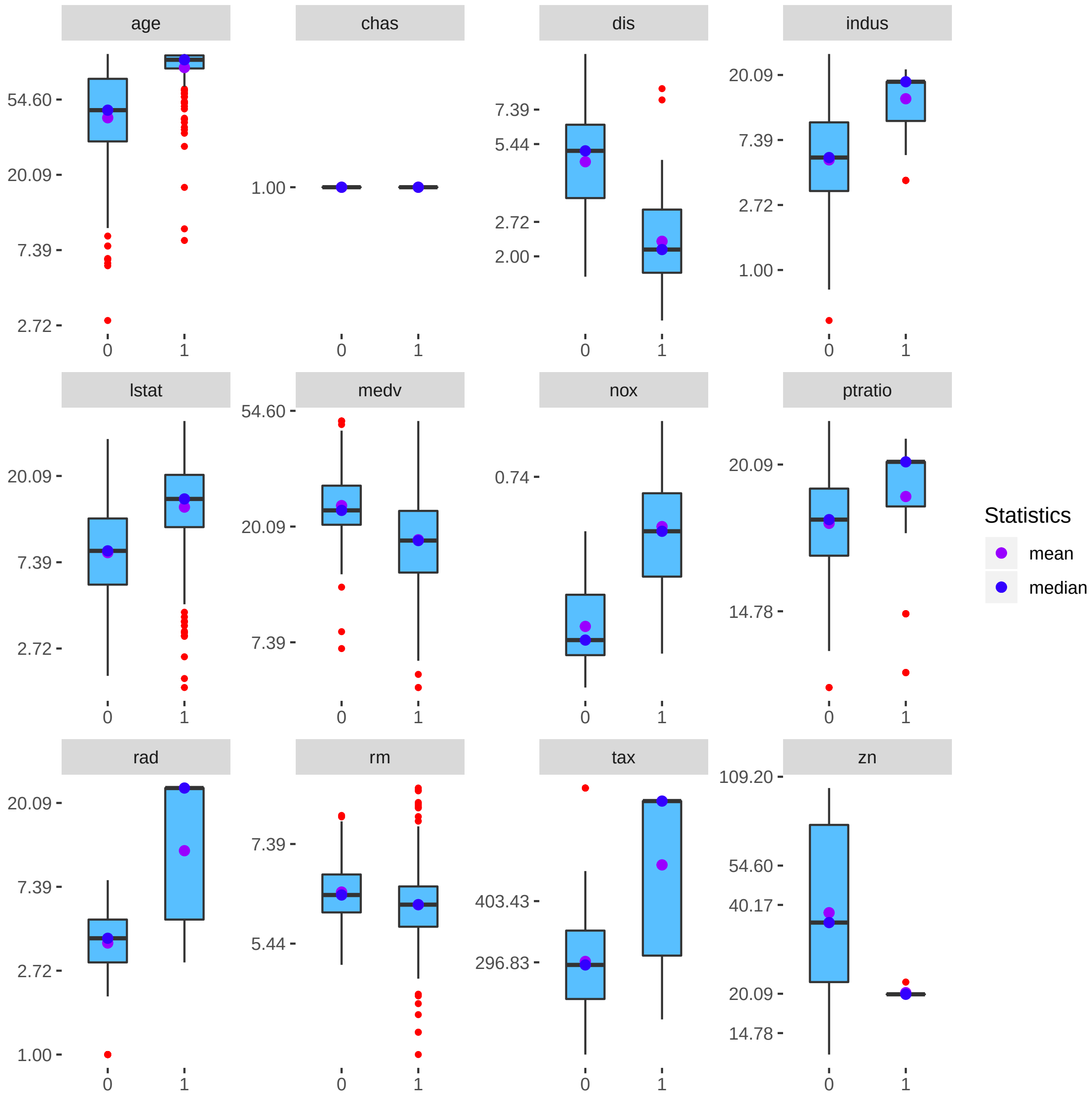
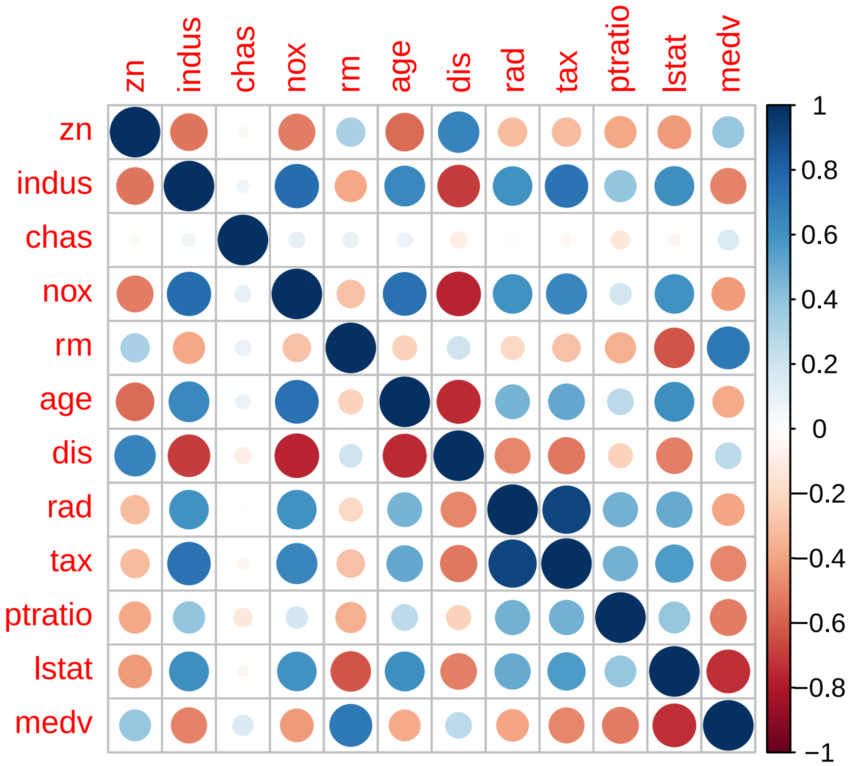


Figure 6: Relationships between natural log transformed predictors and the target

2.3 Correlation

An examination of correlation between the explanatory variables reveals the following:

- `indus` (non-retail business acre proportion) is positively correlated with `nox` (pollution concentration, $r = .76$) and `tax` (property tax rate per \$10k, $r = .73$) and is negatively correlated with `dis` (weighted mean distance to employment centers, $r = -.7$)
- `chas` (bordering Charles river) correlated with `nox` ($r = .97$) and `rm` (average rooms per dwelling, $r = .91$) and `age` (proportion of pre-1940 homes, $r = .79$); and is negatively correlated with `dis` ($r = -.97$)
- `medv` (median value of owner-occupied homes) is correlated with `rm` ($r = .71$); and is negatively correlated with `lstat` (percent lower status population, $r = -.74$)
- `age` is correlated with `nox` ($r = .74$); and is negatively correlated with `dis` ($r = -.75$)
- `rad` (highway access index) correlated with `tax` ($r = .91$)



2.4 Feature Engineering

In 'A Modern Approach to Regression with R' (page 284), Sheather quotes Cook and Weisberg, suggesting that the best way to determine need for log transformation of skewed predictors is to include both the original and transformed variables in the logistic regression model in order assess their relative contributions directly and prune accordingly

Reexamining the histograms of the predictor distributions above reveals that:

- `age` is left-skewed
- `dis` is right-skewed, and `zn` is extremely so
- `nox` is right-skewed and platykurtic (thin-tailed)
- `rad` and `tax` seem to have normal distributions, with large numbers of outliers at particular levels

Table 3: Correlation between predictors

	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	lstat	medv
zn	1.00	-0.54	-0.04	-0.52	0.32	-0.57	0.66	-0.32	-0.32	-0.39	-0.43	0.38
indus	-0.54	1.00	0.06	0.76	-0.39	0.64	-0.70	0.60	0.73	0.39	0.61	-0.50
chas	-0.04	0.06	1.00	0.10	0.09	0.08	-0.10	-0.02	-0.05	-0.13	-0.05	0.16
nox	-0.52	0.76	0.10	1.00	-0.30	0.74	-0.77	0.60	0.65	0.18	0.60	-0.43
rm	0.32	-0.39	0.09	-0.30	1.00	-0.23	0.20	-0.21	-0.30	-0.36	-0.63	0.71
age	-0.57	0.64	0.08	0.74	-0.23	1.00	-0.75	0.46	0.51	0.26	0.61	-0.38
dis	0.66	-0.70	-0.10	-0.77	0.20	-0.75	1.00	-0.49	-0.53	-0.23	-0.51	0.26
rad	-0.32	0.60	-0.02	0.60	-0.21	0.46	-0.49	1.00	0.91	0.47	0.50	-0.40
tax	-0.32	0.73	-0.05	0.65	-0.30	0.51	-0.53	0.91	1.00	0.47	0.56	-0.49
ptratio	-0.39	0.39	-0.13	0.18	-0.36	0.26	-0.23	0.47	0.47	1.00	0.38	-0.52
lstat	-0.43	0.61	-0.05	0.60	-0.63	0.61	-0.51	0.50	0.56	0.38	1.00	-0.74
medv	0.38	-0.50	0.16	-0.43	0.71	-0.38	0.26	-0.40	-0.49	-0.52	-0.74	1.00

- `indus` and `ptratio` reveal peculiar skew, with incidences at particular high level, perhaps due to regulation or infrastructure requirements

We include log transforms of `age`, `dis`, `nox`, `rad`, `tax`, `indus`, and `ptratio` in the dataset for evaluation in models.

3 BUILD MODELS

3.1 Model 1 - Base Model

The First model is a binary logistic model including all the explanatory variables. The data is centered and scaled based on the mean and standard deviation of the variables.

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8464  -0.1445  -0.0017   0.0029   3.4665
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.3290     0.7195   3.237  0.00121 **
## zn           -1.5408     0.8097  -1.903  0.05706 .
## indus        -0.4423     0.3260  -1.357  0.17485
## chas          0.2339     0.1940   1.205  0.22803
## nox           5.7309     0.9254   6.193 5.90e-10 ***
## rm           -0.4141     0.5095  -0.813  0.41637
## age           0.9683     0.3912   2.475  0.01333 *
## dis           1.5563     0.4852   3.208  0.00134 **
## rad           5.7880     1.4171   4.084 4.42e-05 ***
## tax          -1.0362     0.4961  -2.089  0.03674 *
## ptratio       0.8844     0.2782   3.179  0.00148 **
## lstat         0.3258     0.3838   0.849  0.39608
## medv          1.6708     0.6310   2.648  0.00810 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 192.05  on 453  degrees of freedom
## AIC: 218.05
##
## Number of Fisher Scoring iterations: 9
```

The residual deviance is 192.05 and the AIC is 218.05. We will consider this as the baseline for all models.

$$\begin{aligned}\widehat{y} =& 2.33 - 1.54\texttt{zn} - 0.44\texttt{indus} + 0.23\texttt{chas} + 5.73\texttt{nox} - 0.41\texttt{rm} + 0.97\texttt{age} \\ &+ 1.55\texttt{dis} + 5.79\texttt{rad} - 1.04\texttt{tax} + 0.88\texttt{ptratio} + 0.33\texttt{lstat} + 1.67\texttt{medv}\end{aligned}$$

3.1.1 Variance Inflation Factors

The review of the VIF output suggests that some variables are highly collinear and may not be necessary to build a model.

Table 4: Variance Inflation Factors for Model 1

	x
zn	304.88578
indus	49.42303
chas	17.50400
nox	398.18024
rm	120.70934
age	71.17422
dis	109.46005
rad	933.83411
tax	114.44284
ptratio	35.98375
lstat	68.51319
medv	185.15264

Table 5: Area Under the Curve

x
0.9737623

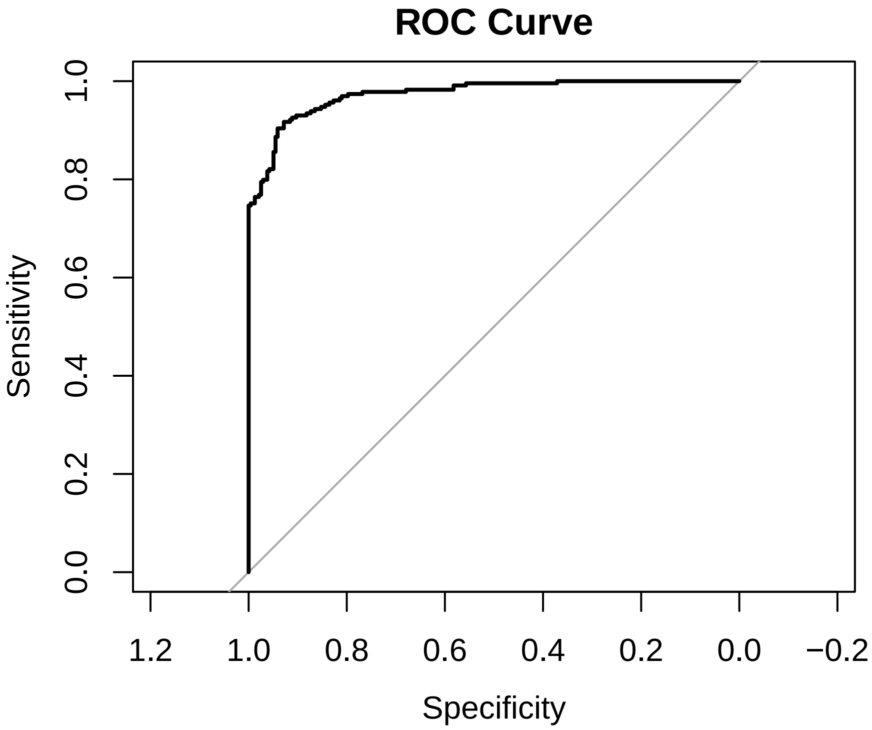


Figure 7: Model 1 ROC Curve

3.2 Model 2 - Log Transform Skewed Predictors and Automated Selection Tools

The second model is a binary logistic model including all the explanatory variables plus log transformations of our skewed variables `age`, `dis`, `nox`, `rad`, `tax`, `indus`, and `ptratio` as recommended by Sheather in 'A Modern Approach to Regression with R'.

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + lstat + medv + log(age) + log(dis) +
##     log(nox) + log(rad) + log(tax) + log(indus) + log(ptratio),
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2967  -0.1011   0.0000   0.0313   4.1276
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -322.86829  145.88834  -2.213 0.026889 *
## zn             -0.05411    0.05837  -0.927 0.353918
## indus           0.37719    0.28387   1.329 0.183934
## chas           -0.51021    0.96978  -0.526 0.598811
## nox           230.71475  124.56195   1.852 0.063996 .
## rm             -1.72607    1.00785  -1.713 0.086782 .
## age             0.09993    0.03738   2.673 0.007514 **
## dis            -2.74240    1.04324  -2.629 0.008570 **
## rad             1.53639    0.39204   3.919 8.89e-05 ***
## tax            -0.19727    0.05496  -3.589 0.000332 ***
## ptratio         5.30769    2.25161   2.357 0.018409 *
## lstat           0.03089    0.05973   0.517 0.605064
## medv            0.29117    0.10617   2.742 0.006100 **
## log(age)       -2.96708    1.47432  -2.013 0.044167 *
## log(dis)       13.41098    4.44624   3.016 0.002559 **
## log(nox)      -98.55289   64.18111  -1.536 0.124650
## log(rad)       -1.31995    1.52479  -0.866 0.386675
## log(tax)       58.71942   16.96167   3.462 0.000536 ***
## log(indus)     -1.54965    2.85552  -0.543 0.587346
## log(ptratio)  -81.99032   39.96120  -2.052 0.040194 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 133.33  on 446  degrees of freedom
## AIC: 173.33
##
## Number of Fisher Scoring iterations: 10
```

3.2.1 Refining with the step function

Using the step function to try to refine this model by removing unnecessary variables leaves us with the following model.

```
##
## Call:
## glm(formula = target ~ indus + nox + rm + age + dis + rad + tax +
##     ptratio + medv + log(age) + log(dis) + log(nox) + log(tax) +
##     log(ptratio), family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3042  -0.1025  -0.0001   0.0372   3.9003
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -265.31253  134.49565  -1.973 0.048536 *
## indus           0.24065    0.10077   2.388 0.016935 *
## nox           205.55579  114.50617   1.795 0.072630 .
## rm             -1.74440    0.84294  -2.069 0.038506 *
## age             0.11122    0.03270   3.401 0.000671 ***
## dis            -2.98947    0.88888  -3.363 0.000770 ***
## rad             1.27330    0.24172   5.268 1.38e-07 ***
## tax            -0.17066    0.04136  -4.126 3.69e-05 ***
## ptratio         5.19720    2.21121   2.350 0.018754 *
## medv            0.27641    0.10030   2.756 0.005857 **
## log(age)       -3.51456    1.29707  -2.710 0.006736 **
## log(dis)       14.09302    3.98914   3.533 0.000411 ***
## log(nox)      -83.67304   58.53436  -1.429 0.152870
## log(tax)       49.92898   12.07873   4.134 3.57e-05 ***
## log(ptratio)  -78.98569   39.06053  -2.022 0.043162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 135.07  on 451  degrees of freedom
## AIC: 165.07
##
## Number of Fisher Scoring iterations: 9
```

3.2.2 Refining further with backward elimination

Since `log(nox)` has such a high p-value removing it from the model doesn't change the model much.

```
##
## Call:
## glm(formula = target ~ indus + nox + rm + age + dis + rad + tax +
##     ptratio + medv + log(age) + log(dis) + log(tax) + log(ptratio),
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3318  -0.0672   0.0000   0.0618   4.0452
##
## Coefficients:
```

```
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -111.78831   78.83784  -1.418 0.156205
## indus           0.27808    0.09432   2.948 0.003196 **
## nox            44.58309    9.94220   4.484 7.32e-06 ***
## rm             -1.53536    0.80944  -1.897 0.057853 .
## age             0.11167    0.03226   3.461 0.000538 ***
## dis            -2.30410    0.74724  -3.083 0.002046 **
## rad             1.31663    0.24894   5.289 1.23e-07 ***
## tax            -0.17956    0.04301  -4.175 2.98e-05 ***
## ptratio         5.87010    2.11023   2.782 0.005407 **
## medv            0.24697    0.09578   2.578 0.009925 **
## log(age)       -4.01196    1.26801  -3.164 0.001556 **
## log(dis)       11.12110    3.33573   3.334 0.000856 ***
## log(tax)       52.55106   12.50157   4.204 2.63e-05 ***
## log(ptratio)  -91.86014   37.01959  -2.481 0.013087 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.88  on 465  degrees of freedom
## Residual deviance: 137.14  on 452  degrees of freedom
## AIC: 165.14
##
## Number of Fisher Scoring iterations: 9
```

So our final equation for the second model is:

$$\begin{aligned}\widehat{y} =& -111.79 + 0.28\texttt{indus} + 44.58\texttt{nox} - 1.54\texttt{rm} + 0.11\texttt{age}\\ &- 2.30\texttt{dis} + 1.32\texttt{rad} - 0.18\texttt{tax} + 5.87\texttt{ptratio} + 0.24\texttt{medv}\\ &- 4.01\texttt{log(age)} + 11.12\texttt{log(dis)} + 52.55\texttt{log(tax)} - 91.86\texttt{log(ptratio)}\end{aligned}$$

Table 6: Area Under the Curve

x
0.9866232

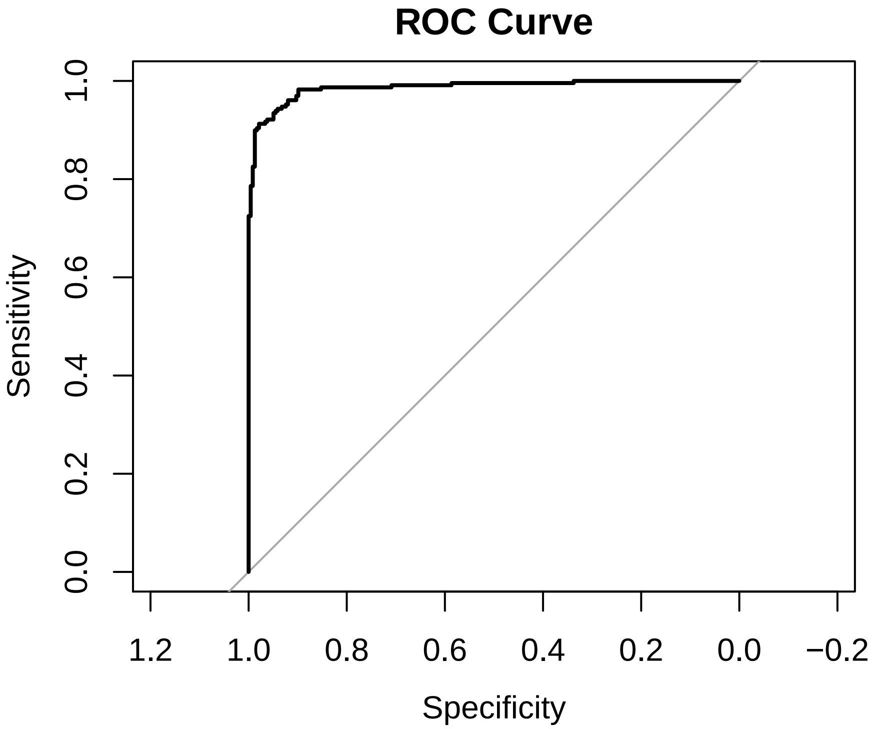


Figure 8: Model 2 ROC Curve

3.3 Model 3 - Examine all possible interactions

For Model 3 we started with a strategy suggested by Faraway in ‘Extending the Linear Model with R’ that adds all possible interactions between the predictor variables in addition to the full set of predictors then uses the step function to remove unnecessary variables or interactions. The resulting model is as follows:

```
##
## Call:
## glm(formula = target ~ zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + lstat + medv + zn:age + zn:tax + zn:ptratio +
##     zn:lstat + indus:chas + indus:rad + indus:ptratio + indus:medv +
##     nox:age + nox:tax + nox:ptratio + nox:lstat + nox:medv +
##     rm:age + age:tax + age:ptratio + dis:tax + dis:ptratio +
##     dis:lstat + dis:medv + rad:tax + tax:medv + lstat:medv, family = binomial,
##     data = train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.069e-03  -2.000e-08  -2.000e-08   2.000e-08   1.150e-03
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.762e+03  7.586e+05   0.005    0.996
## zn             3.308e+02  1.018e+04   0.033    0.974
## indus         -8.252e+02  1.830e+04  -0.045    0.964
## chas           5.607e+02  3.014e+04   0.019    0.985
## nox            6.112e+04  1.331e+06   0.046    0.963
## rm            -5.252e+02  1.738e+04  -0.030    0.976
## age           -8.523e+01  2.926e+03  -0.029    0.977
## dis           -4.914e+03  1.426e+05  -0.034    0.973
## rad            1.481e+02  2.866e+03   0.052    0.959
## tax           -7.452e+01  1.585e+03  -0.047    0.962
## ptratio        1.110e+03  2.519e+04   0.044    0.965
## lstat          7.347e+02  1.406e+04   0.052    0.958
## medv          -9.237e+02  2.093e+04  -0.044    0.965
## zn:age        -1.623e+00  5.060e+01  -0.032    0.974
## zn:tax        -3.277e-01  1.264e+01  -0.026    0.979
## zn:ptratio    -1.239e+01  3.604e+02  -0.034    0.973
## zn:lstat       6.001e+00  2.359e+02   0.025    0.980
## indus:chas    -8.051e+01  2.327e+03  -0.035    0.972
## indus:rad      4.608e+01  1.174e+03   0.039    0.969
## indus:ptratio  4.202e+01  9.357e+02   0.045    0.964
## indus:medv    -4.728e+00  2.070e+02  -0.023    0.982
## nox:age       -1.223e+02  2.696e+03  -0.045    0.964
## nox:tax        4.239e+01  1.160e+03   0.037    0.971
## nox:ptratio   -3.981e+03  7.200e+04  -0.055    0.956
## nox:lstat     -8.463e+02  1.671e+04  -0.051    0.960
## nox:medv       1.192e+03  3.257e+04   0.037    0.971
## rm:age         6.344e+00  1.887e+02   0.034    0.973
## age:tax        1.547e-01  3.437e+00   0.045    0.964
## age:ptratio    3.565e+00  9.585e+01   0.037    0.970
## dis:tax        6.290e+00  1.485e+02   0.042    0.966
## dis:ptratio    1.405e+02  5.035e+03   0.028    0.978
## dis:lstat     -2.926e+01  7.928e+02  -0.037    0.971
```

Table 7: Area Under the Curve

x
1

```
## dis:medv       3.558e+01  7.316e+02   0.049    0.961
## rad:tax       -9.327e-01  2.180e+01  -0.043    0.966
## tax:medv       9.764e-01  1.759e+01   0.055    0.956
## lstat:medv    -8.433e+00  1.604e+02  -0.053    0.958
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6.4588e+02  on 465  degrees of freedom
## Residual deviance: 1.4059e-05  on 430  degrees of freedom
## AIC: 72
##
## Number of Fisher Scoring iterations: 25
```

Although we got the best AIC (72) and residual deviance numbers with this model, the coefficients are all insanely large and so are the p-values with not a single one showing any significance. This is a good example of an extremely over-fitted model. It probably models our training data perfectly, but would perform very poorly at predicting our test data.

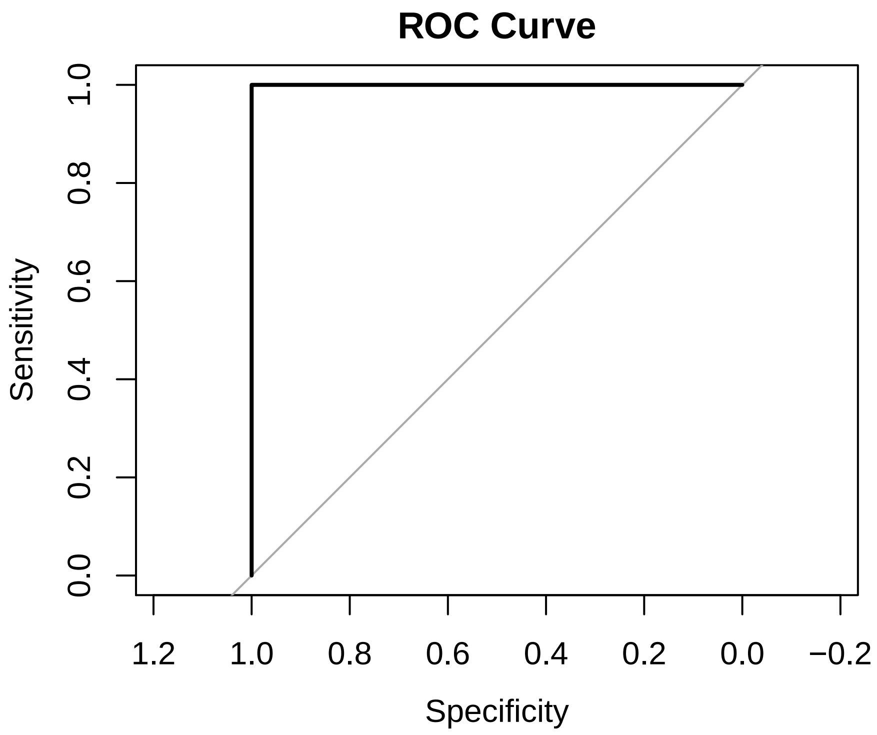


Figure 9: Model 3 ROC Curve

3.4 Model 4 - Segmented/Piecewise Regression

Model 4 uses a broken stick approach based on the evidence we saw in the predictor distributions that indicated that splitting `rad` at a value of 5 would separate most 0's from 1's in our target. We also included the log transformed variables that remained at the end of the model 2 selection process and interaction terms for the variables that showed the greatest correlation in our correlation plot which we will later narrow down using the step function.

```
## 
## Call:
## glm(formula = target ~ (less_than_five(rad) + five_and_over(rad)) + 
##     zn + indus + chas + nox + rm + age + dis + tax + ptratio + 
##     lstat + medv + log(age) + log(dis) + log(tax) + log(ptratio) + 
##     indus:nox + indus:dis + indus:tax + nox:age + nox:dis + rm:medv + 
##     dis:age, family = binomial, data = train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.5048  -0.0820  -0.0001   0.0258   4.5052  
## 
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         -1.101e+02  1.906e+02  -0.578 0.563501    
## less_than_five(rad)  2.891e+00  6.343e-01   4.557 5.19e-06 ***
## five_and_over(rad)   1.818e+00  4.202e-01   4.327 1.51e-05 ***
## zn                  -1.396e-01  7.114e-02  -1.962 0.049720 *  
## indus                9.711e-01  1.918e+00   0.506 0.612570    
## chas                -8.803e-01  1.223e+00  -0.720 0.471602    
## nox                  1.722e+02  9.091e+01   1.894 0.058164 .  
## rm                  -3.458e+00  2.091e+00  -1.653 0.098250 .  
## age                  5.823e-01  3.464e-01   1.681 0.092773 .  
## dis                  6.672e+00  5.073e+00   1.315 0.188489    
## tax                 -2.977e-01  1.003e-01  -2.969 0.002989 ** 
## ptratio              1.155e+01  5.244e+00   2.202 0.027675 *  
## lstat                3.014e-02  9.026e-02   0.334 0.738419    
## medv                -2.304e-01  5.032e-01  -0.458 0.647125    
## log(age)            -2.621e+00  2.137e+00  -1.227 0.219971    
## log(dis)             2.914e+01  1.281e+01   2.274 0.022983 *  
## log(tax)             8.400e+01  2.486e+01   3.378 0.000729 ***
## log(ptratio)        -2.034e+02  9.397e+01  -2.164 0.030437 *  
## indus:nox           -1.987e+00  3.631e+00  -0.547 0.584173    
## indus:dis           -2.100e-01  1.570e-01  -1.338 0.180967    
## indus:tax            2.349e-03  2.840e-03   0.827 0.408201    
## nox:age             -7.641e-01  5.563e-01  -1.374 0.169548    
## nox:dis             -2.094e+01  1.357e+01  -1.544 0.122644    
## rm:medv              6.909e-02  6.659e-02   1.038 0.299485    
## age:dis             -2.614e-02  1.601e-02  -1.632 0.102662    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 645.876  on 465  degrees of freedom
## Residual deviance:  88.417  on 441  degrees of freedom
## AIC: 138.42
##
## Number of Fisher Scoring iterations: 10
```

Clearly separating the data based on splitting the `rad` variable at a value of 5 showed a great improvement in the model. The two `rad` predictors for values 5 and above or less than 5 are by far the two most significant predictors in our model. There are still a lot of variables with very low significance though and the model could definitely use some refinement.

3.4.1 Backward Elimination vs. Forward Selection

We tried using using the step function for both forward selection and backward elimination and found that the backward elimination process resulted in the better model. There were still some variables that appeared to be adding little value to the model however, so we removed `dis`, then `medv`, then the intercept resulting in the final model below.

3.4.2 Final Model

```
##
## Call:
## glm(formula = target ~ less_than_five(rad) + five_and_over(rad) +
##     zn + indus + nox + rm + age + tax + ptratio + log(dis) +
##     log(tax) + log(ptratio) + indus:dis + rm:medv + age:dis +
##     0, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4422  -0.0681   0.0000   0.0209   4.4906
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## less_than_five(rad)    3.00881    0.51465   5.846 5.02e-09 ***
## five_and_over(rad)     1.79754    0.33931   5.298 1.17e-07 ***
## zn                    -0.14902    0.05252  -2.837 0.004547 **
## indus                  0.52513    0.27310   1.923 0.054499 .
## nox                   32.54513   11.57059   2.813 0.004912 **
## rm                    -3.59394    1.46963  -2.445 0.014466 *
## age                    0.10112    0.04520   2.237 0.025284 *
## tax                   -0.22145    0.05670  -3.906 9.40e-05 ***
## ptratio               10.86558    2.61887   4.149 3.34e-05 ***
## log(dis)              11.52180    4.10186   2.809 0.004971 **
## log(tax)              69.01932   18.06280   3.821 0.000133 ***
## log(ptratio)        -190.89982   46.28125  -4.125 3.71e-05 ***
## indus:dis             -0.17764    0.08886  -1.999 0.045599 *
## rm:medv                0.04771    0.01731   2.757 0.005842 **
## age:dis               -0.01901    0.01045  -1.819 0.068964 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 646.013  on 466  degrees of freedom
```

Table 8: Area Under the Curve

x
0.9921508

```
## Residual deviance:  94.393  on 451  degrees of freedom
## AIC: 124.39
##
## Number of Fisher Scoring iterations: 10
```

So our final equation for the fourth model is:

$$\begin{aligned}
\widehat{y} &= 0 + 3.01\texttt{rad (less than 5)} + 1.80\texttt{rad (five and over)} - 0.15\texttt{zn} \\
&+ 0.53\texttt{indus} + 32.55\texttt{nox} - 3.59\texttt{rm} + 0.10\texttt{age} - 0.22\texttt{tax} + 10.87\texttt{ptratio} + 11.52\texttt{log(dis)} \\
&+ 69.02\texttt{log(tax)} - 190.90\texttt{log(ptratio)} - 0.18\texttt{indus:dis} + 0.04\texttt{rm:medv} + -0.02\texttt{age:dis}
\end{aligned}$$

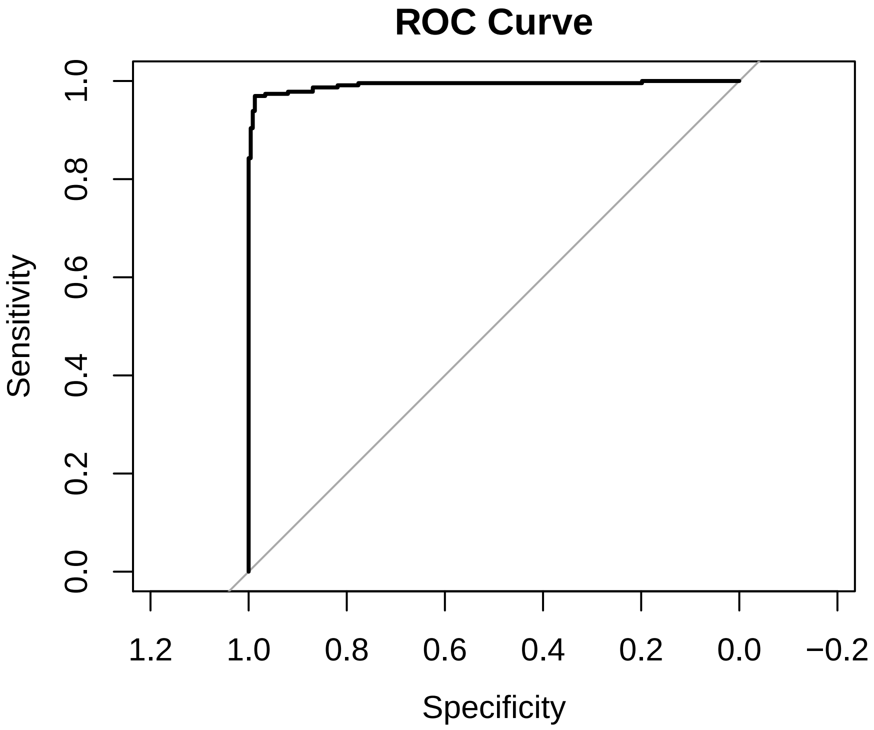


Figure 10: Model 4 ROC Curve

3.4.3 Marginal Model Plots

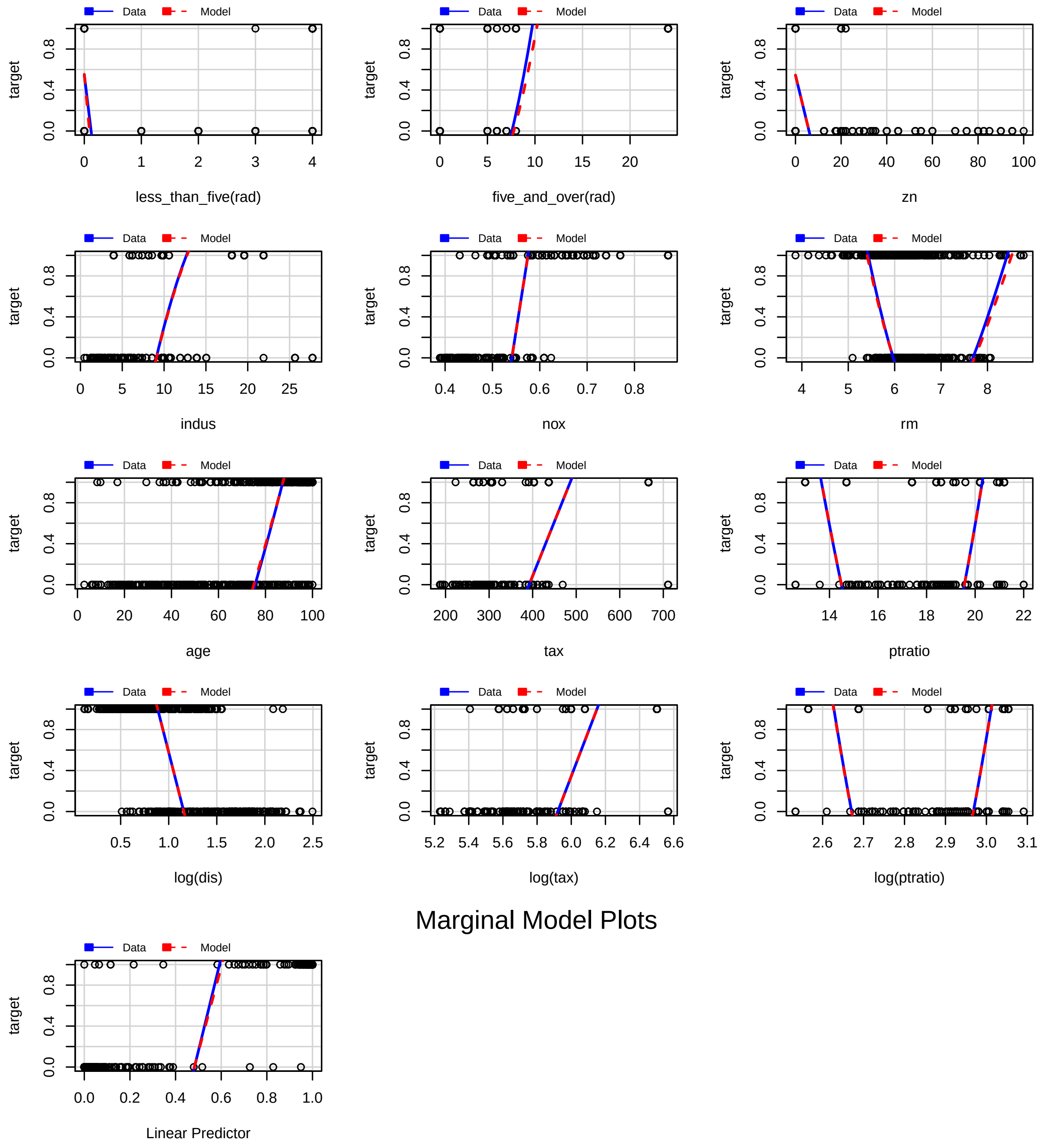


Table 9: Confusion Matrix Summary Statistics

	Sensitivity	Specificity	Precision	Recall	F1
Model.1	0.9282700	0.9039301	0.9090909	0.9282700	0.9185804
Model.2	0.9662447	0.9126638	0.9196787	0.9662447	0.9423868
Model.3	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Model.4	0.9873418	0.9650655	0.9669421	0.9873418	0.9770355

Table 10: Pseudo R2

	llh	llhNull	G2	McFadden	r2ML	r2CU
Model.1	-96.023459	-322.9379	453.8289	0.7026566	0.6223856	0.8299290
Model.2	-68.568641	-322.9379	508.7385	0.7876724	0.6643592	0.8858993
Model.3	-0.000007	-322.9379	645.8758	1.0000000	0.7499263	1.0000000
Model.4	-47.196447	-322.9379	551.4829	0.8538529	0.6937764	0.9251261

4 SELECT MODELS

The four models were explored in order to determine the best way to determine whether or not a neighborhood's crime rate was above or below the median crime rate. It has been established that the most efficient model was the fourth model, with the first model being somewhat efficient, and the third over-fitted model being least efficient.

4.1 Pseudo R2

There is no R^2 for logistic regression to further evaluate, however, there is an alternative called $pseudoR^2$ terms that can be used for evaluation.

All of these measures and especially McFadden support the anova test's conclusion that model 4.

4.2 Summary diagnostic plots

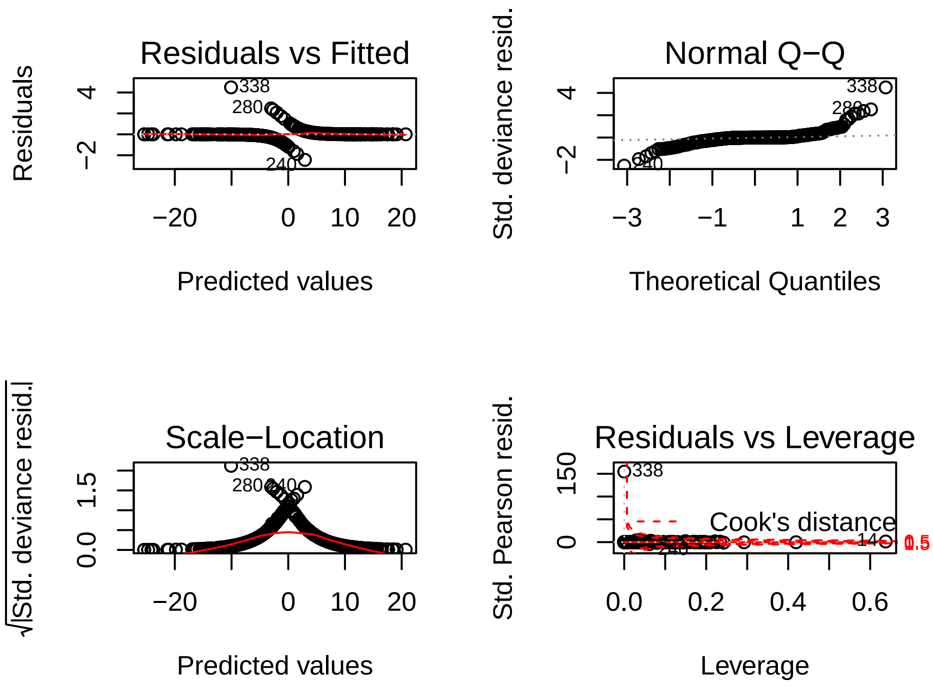


Figure 11: Model 4 Summary diagnostic plots

5 Appendix

The appendix is available as script.R file in `project3_crime` folder.

https://github.com/betsyrosalen/DATA_621_Business_Analyt_and_Data_Mining

```
# Load libs

if (!require('car')) (install.packages('car'))
if (!require('data.table')) (install.packages('data.table'))
if (!require('ggplot2')) (install.packages('ggplot2'))
if (!require('caret')) (install.packages('caret'))
if (!require('corrplot')) (install.packages('corrplot'))
if (!require('data.table')) (install.packages('data.table'))
if (!require('DataExplorer')) (install.packages('DataExplorer'))
if (!require('faraway')) (install.packages('faraway'))
if (!require('formattable')) (install.packages('formattable'))
if (!require('gridExtra')) (install.packages('gridExtra'))
if (!require('kableExtra')) (install.packages('kableExtra'))
if (!require('leaps')) (install.packages('leaps'))
if (!require('MASS')) (install.packages('MASS'))
if (!require('nnet')) (install.packages('nnet'))
if (!require('psych')) (install.packages('psych'))
if (!require('pROC')) (install.packages('pROC'))
if (!require('reshape')) (install.packages('reshape'))
if (!require('tidyverse')) (install.packages('tidyverse'))
if (!require('randomForest')) (install.packages('randomForest'))

# load data
train <- read.csv ('https://raw.githubusercontent.com/silverrainb/data621proj3/master/crime-training-dat
test <- read.csv('https://raw.githubusercontent.com/silverrainb/data621proj3/master/crime-evaluation-dat

variable_descriptions <- rbind(c('target','whether the crime rate is above the median crime rate (1) or
                               c('zn','proportion of residential land zoned for large lots (over 25000 s
                               c('indus','proportion of non-retail business acres per suburb','predictor
                               c('chas','a dummy var. for whether the suburb borders the Charles River
                               c('nox','nitrogen oxides concentration (parts per 10 million)','predictor
                               c('rm','average number of rooms per dwelling','predictor' ),
                               c('age','proportion of owner-occupied units built prior to 1940','predict
                               c('dis','weighted mean of distances to five Boston employment centers','p
                               c('rad','index of accessibility to radial highways','predictor'),
                               c('tax','full-value property-tax rate per $10,000','predictor'),
                               c('ptratio','pupil-teacher ratio by town','predictor'),
                               c('black','1000(B_k - 0.63)^2 where B_k is the proportion of blacks by to
                               c('lstat','lower status of the population (percent)','predictor'),
                               c('medv','median value of owner-occupied homes in $1000s','predictor'))
colnames(variable_descriptions) <- c('VARIABLE','DEFINITION','TYPE')

# Summary Statistics
sum_stat <- describe(train)[,c(2,8,3,5,9,4)]

# Shape of Predictor Distributions
Hist_new <- train %>%
    gather(-target, key = "var", value = "val") %>%
    ggplot(aes(x = val, fill=factor(target))) +
    geom_histogram(position="dodge", bins=10, alpha=0.5) +
    facet_wrap(~ var, scales = "free") +
    scale_fill_manual("target",values = c("#58BFFF", "#3300FF")) +
    xlab("") +
    ylab("") +
    theme(panel.background = element_blank())


# Outliers
boxplot_train <- train[,-13]
boxplot_train$tax <- boxplot_train$tax/10
melt.train <- melt(boxplot_train)

outlier.boxplot <- ggplot(melt.train, aes(variable, value)) +
  geom_boxplot(width=.5, fill="#58BFFF", outlier.colour="red", outlier.size = 1) +
  stat_summary(aes(colour="mean"), fun.y=mean, geom="point",
               size=2, show.legend=TRUE) +
  stat_summary(aes(colour="median"), fun.y=median, geom="point",
               size=2, show.legend=TRUE) +
  coord_flip(ylim = c(0, 110), expand = TRUE) +
  scale_y_continuous(labels = scales::comma,
                     breaks = seq(0, 110, by = 10)) +
  labs(colour="Statistics", x="", y="") +
  scale_colour_manual(values=c("#9900FF", "#3300FF")) +
  theme(panel.background=element_blank(), legend.position="top")


#  Missing Values
na.barplot <- plot_missing(train)

# Boxplots
boxplots <- train %>%
    gather(-target, key = "var", value = "val") %>%
    ggplot(aes(x=factor(target), y=val)) +
    geom_boxplot(width=.5, fill="#58BFFF", outlier.colour="red", outlier.size = 1) +
    stat_summary(aes(colour="mean"), fun.y=mean, geom="point",
                 size=2, show.legend=TRUE) +
    stat_summary(aes(colour="median"), fun.y=median, geom="point",
                 size=2, show.legend=TRUE) +
    facet_wrap(~ var, scales = "free", ncol=4) +
    labs(colour="Statistics", x="", y="") +
    scale_colour_manual(values=c("#9900FF", "#3300FF")) +
    theme(panel.background=element_blank())

# DATA PREPARATION <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

# Correlation

correl2 <- train %>%
  select(-target) %>%
  cor() %>%
  round(2) %>%
  corrplot(method = "circle")

# Our transformation function
scaleFUN <- function(x) sprintf("%.2f", x)

Hist_log_new <- train %>%
    gather(-target, key = "var", value = "val") %>%
    ggplot(aes(x = val, fill=factor(target))) +
    geom_histogram(position="dodge", bins=10, alpha=0.5) +
    facet_wrap(~ var, scales = "free") +
    scale_y_continuous(trans = "log", label=scaleFUN) +
    scale_fill_manual("target",values = c("#58BFFF", "#3300FF")) +
    xlab("") +
    ylab("") +
    theme(panel.background = element_blank())

# Linearity at log10 scale

linearity_log_new <- train %>%
    gather(-target, key = "var", value = "val") %>%
    ggplot(aes(x=factor(target), y=val)) +
    geom_boxplot(width=.5, fill="#58BFFF", outlier.colour="red", outlier.size = 1) +
    scale_y_continuous(trans = "log", label=scaleFUN) +
    stat_summary(aes(colour="mean"), fun.y=mean, geom="point",
                 size=2, show.legend=TRUE) +
    stat_summary(aes(colour="median"), fun.y=median, geom="point",
                 size=2, show.legend=TRUE) +
    facet_wrap(~ var, scales = "free", ncol=4) +
    labs(colour="Statistics", x="", y="") +
    scale_colour_manual(values=c("#9900FF", "#3300FF")) +
    theme(panel.background=element_blank())

# BUILD MODELS <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

## Model 1

## Build the model
model.1 <- glm(target ~ .,
               family = binomial,
               data = train) # + 0 removes intercept
#
mod.1 <- train(target ~., data = train,
                 method = "glm",
                 family = "binomial",
                 preProcess = c("center", "scale")) # center and scale data based on the mean and sd

mod1_summary <- summary(mod.1)

### Model 1 Summary Statistics
pred.1.raw <- predict(mod.1, newdata = train)
pred.1 <- as.factor(ifelse(pred.1.raw < .5, 0, 1))
mod1.conf.mat <- confusionMatrix(pred.1,
                                 as.factor(train$target), mode = "everything")

#=====================================================================================================
## Model 2

## Build the model
model.2.raw <- glm(target ~ zn + indus + chas + nox + rm + age + dis + rad + tax +
                     ptratio + lstat + medv + log(age) + log(dis) + log(nox) +
                     log(rad) + log(tax) + log(indus) + log(ptratio),
               family = binomial,
               data = train)

model.2.step <- step(model.2.raw, trace=FALSE)

model.2 <- glm(target ~ indus + nox + rm + age + dis + rad + tax + ptratio + medv +
                     log(age) + log(dis) + log(tax) + log(ptratio),
                     family = binomial, data = train)

mod2_summary <- summary(model.2)

#marg_mod_plot_2 <- mmps(model.2, layout=c(5,4), key=NULL) # library car

### Model 2 Summary Statistics
pred.2.raw <- predict(model.2, newdata = train)
pred.2 <- as.factor(ifelse(pred.2.raw < .5, 0, 1))
mod2.conf.mat <- confusionMatrix(pred.2,
                                 as.factor(train$target), mode = "everything")

#======================================================================================================

## Model 3

#big_mod5 <- glm(target ~ (zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + lstat + medv
#               data = train, family = binomial)

#small_mod5 <- step(big_mod5, trace=FALSE)

# The above code is VERY computationally expensive
# Here's the result so it doesn't need to be run again.
model.5 <-  glm(formula = target ~ zn + indus + chas + nox + rm + age + dis +
                    rad + tax + ptratio + lstat + medv + zn:age + zn:tax + zn:ptratio +
                    zn:lstat + indus:chas + indus:rad + indus:ptratio + indus:medv +
                    nox:age + nox:tax + nox:ptratio + nox:lstat + nox:medv +
                    rm:age + age:tax + age:ptratio + dis:tax + dis:ptratio+
                    dis:lstat + dis:medv + rad:tax + tax:medv + lstat:medv, family = binomial,
                    data = train)

mod5_summary <- summary(model.5)

#resid_plot_5 <- residual.plots(model.5, exclude = 4, layout = c(2, 2)) # library car

#marg_mod_plot_5 <- mmps(model.5, span = 3/4, layout = c(2, 2)) # library car

### Model 5 Summary Statistics
pred.5.raw <- predict(model.5, newdata = train)
pred.5 <- as.factor(ifelse(pred.5.raw < .5, 0, 1))
mod5.conf.mat <- confusionMatrix(pred.5, as.factor(train$target), mode = "everything")


#======================================================================================================
## Model 4

## Build the model
less_than_five <- function(x) ifelse(x < 5, x, 0)
five_and_over <- function(x) ifelse(x >= 5, x, 0)

model.6.raw <- glm(target ~ (less_than_five(rad) + five_and_over(rad)) + zn + indus + chas + nox +
                       rm + age + dis + tax + ptratio + lstat + medv + log(age) +
                       log(dis) + log(tax) + log(ptratio) + indus:nox + indus:dis +
                       indus:tax+ nox:age + nox:dis + rm:medv + dis:age,
                   family = binomial,
                   data = train)

backward.mod <- step(model.6.raw, direction = "backward", trace=FALSE)
backward_sum <- summary(backward.mod)

#forward.mod <- step(model.6.raw, direction = "forward", trace=FALSE)
#forward_sum <- summary(forward.mod)

model.6 <- glm(target ~ less_than_five(rad) + five_and_over(rad) +
                    zn + indus + nox + rm + age + tax + ptratio +
                    log(dis) + log(tax) + log(ptratio) + indus:dis + rm:medv +
                    age:dis + 0, family = binomial, data = train) # + 0 removes intercept

mod6_summary <- summary(model.6)

### Model 6 Summary Statistics
pred.6.raw <- predict(model.6, newdata = train)
pred.6 <- as.factor(ifelse(pred.6.raw < .5, 0, 1))
mod6.conf.mat <- confusionMatrix(pred.6, as.factor(train$target), mode = "everything")

mod.6 <- train(target ~ less_than_five(rad) + five_and_over(rad) +
                 zn + indus + nox + rm + age + tax + ptratio +
                 log(dis) + log(tax) + log(ptratio) + indus:dis + rm:medv +
                 age:dis + 0,
               family = binomial,
               data = train,
               method = 'glm') # + 0 removes intercept

#======================================================================================================

## Model Evaluations

eval_mods <- data.frame(mod1.conf.mat$byClass,
                   mod2.conf.mat$byClass,
                   mod5.conf.mat$byClass,
                   mod6.conf.mat$byClass) # add additional model stats

eval_mods <- data.frame(t(eval_mods))
row.names(eval_mods) <- c("Model.1", "Model.2", "Model.3", "Model.4") # add additional models

eval_mods <- dplyr::select(eval_mods, Sensitivity, Specificity, Precision, Recall, F1)


# SELECT MODELS <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

#Pseudo R2

pseudo.r2 <- data.frame(pscl::pR2(model.1),
                        pscl::pR2(model.2),
                        pscl::pR2(model.5),
                        pscl::pR2(model.6))

pseudo.r2 <- data.frame(t(pseudo.r2))

row.names(pseudo.r2) <- c("Model.1", "Model.2", "Model.3", "Model.4")
```